

ON THE IMPLEMENTATION OF EXPONENTIAL METHODS FOR SEMILINEAR PARABOLIC EQUATIONS

MARÍA LÓPEZ-FERNÁNDEZ [†]

Abstract. The time integration of semilinear parabolic problems by exponential methods of different kinds is considered. A new algorithm for the implementation of these methods is proposed. The algorithm evaluates the operators required by the exponential methods by means of a quadrature formula that converges like $O(e^{-cK/\ln K})$, with K the number of quadrature nodes. The algorithm allows also the evaluation of the associated scalar mappings and in this case the quadrature converges like $O(e^{-cK})$. The technique is based on the numerical inversion of sectorial Laplace transforms. Several numerical illustrations are provided to test the algorithm.

Key words. exponential methods, numerical inverse Laplace transform, semilinear parabolic equations.

AMS subject classifications. 65M70, 65R10.

1. Introduction. The good numerical results obtained from the application of exponential methods to the time integration of stiff semilinear problems, have motivated much interest on these kind of methods during the last years, see for instance [1, 2, 3, 9, 10, 11]. The problems under consideration can be written in the abstract format

$$u'(t) = Au(t) + f(t, u(t)), \quad u(0) = u_0, \quad 0 \leq t \leq T, \quad (1.1)$$

where A is a linear operator representing the highest order differential terms and f is a lower order nonlinear operator. The solution to the initial value problem (1.1) is then given by the variation of constants formula and most of the exponential methods considered in the literature are constructed from this representation of the solution.

Let us consider for instance the family of multistep exponential methods developed in [2] and briefly reviewed in Section 2.1. Given a stepsize $h > 0$, $n \geq 0$ and approximations $u_{n+j} \approx u(t_{n+j})$, $t_{n+j} = (n+j)h$, $0 \leq j \leq k-1$, the k -step method approximates the solution u of (1.1) at $t_{n+k} = (n+k)h$ by

$$u_{n+k} = \phi_0(k, hA)u_n + h \sum_{j=0}^{k-1} \phi_{j+1}(k, hA) \Delta^j f_n, \quad (1.2)$$

where $f_n = f(t_n, u_n)$, Δ denotes the standard forward difference operator, and for $\lambda \in \mathbb{C}$ and $k \geq 1$,

$$\phi_0(k, \lambda) = e^{k\lambda}, \quad \phi_j(k, \lambda) = \int_0^k e^{(k-\sigma)\lambda} \binom{\sigma}{j-1} d\sigma, \quad 1 \leq j \leq k. \quad (1.3)$$

As we can see in (1.2), these methods require the evaluation of $\phi_j(k, hA)$, $0 \leq j \leq k$, for $\phi_j(k, \lambda)$ defined in (1.3). This is in fact the main difficulty in the implementation of the methods in (1.2) and, in general, of exponential methods, since they typically

[†]Departamento de Matemáticas, Universidad Autónoma de Madrid, Madrid, Spain. E-mail: maria.lopez@uam.es. Supported by DGI-MCYT under projects MTM 2005-00714 and MTM 2007-63257, cofinanced by FEDER funds, and the SIMUMAT project S-0505/ESP/0158 of the Council of Education of the Regional Government of Madrid (Spain).
July 28, 2008.

require the evaluation of vector-valued mappings $\phi(hA)$, with h the time step in the discretization and either

$$\phi(\lambda) = e^{m\lambda}, \quad \lambda \in \mathbb{C}, \quad (1.4)$$

or

$$\phi(\lambda) = \int_0^m e^{(m-\sigma)\lambda} p(\sigma) d\sigma, \quad (1.5)$$

with m an integer and $p(\sigma)$ a polynomial. The values of m and p in (1.5) depend on the method. For instance, in the case of the methods in (1.2), it is clear from (1.3) that $m = k$, the number of steps of the method, and

$$p(\sigma) = \binom{\sigma}{j-1} = \frac{\sigma(\sigma-1)\dots(\sigma-j+2)}{(j-1)!}, \quad 1 \leq j \leq k.$$

For the explicit exponential Runge–Kutta methods constructed in [10] it is $m = 1$ and

$$p(\sigma) = \frac{\sigma^{k-1}}{(k-1)!}, \quad k \geq 1,$$

as we can see in Section 2.2, (2.7).

In the present paper we propose a way to evaluate the operators $\phi(hA)$ in (1.4) and (1.5) when A in (1.1) is the infinitesimal generator of an analytic semigroup in a Banach space X . Thus, we will assume that $A : D(A) \subset X \rightarrow X$ is sectorial, i.e., A is a densely defined and closed linear operator on X and there exist constants $M > 0$, $\gamma \in \mathbb{R}$, and an angle $0 < \delta < \frac{\pi}{2}$, such that the resolvent fulfils

$$\|(zI - A)^{-1}\| \leq \frac{M}{|z - \gamma|}, \quad \text{for } |\arg(z - \gamma)| < \pi - \delta. \quad (1.6)$$

As a side product we also obtain an accurate way to evaluate the mappings $\phi(\lambda)$ in (1.5) at the scalar level, which is itself a well-known problem in numerical analysis. This is exemplified in [11] with the mapping

$$\varphi(\lambda) = \frac{e^\lambda - 1}{\lambda}, \quad (1.7)$$

which is required for instance by the exponential Runge–Kutta methods of [10]. The evaluation of φ for small λ by using formula (1.7) suffers from cancellation error. On the other hand, the use of a truncated Taylor expansion only works well for small enough λ . This implies that for some intermediate values of λ it is not very much clear how to choose the proper formula and moreover both of them could lose accuracy. For λ inside a sector $|\arg(\lambda - \gamma)| \geq \pi - \delta$, these difficulties can be overcome by writing φ in the format (1.5), with $m = 1$ and $p(\sigma) = 1$. By doing so, we will be able to evaluate $\varphi(\lambda)$, independently of the size of λ , by using essentially the same technique developed in principle to evaluate the vector-valued mapping $\varphi(hA)$.

In the recent literature, several alternatives have been proposed to evaluate the required operators $\phi(hA)$ or alternatively their action on given vector, based on a Krylov approach [7, 8], on Padé approximants [1] or on a suitable contour integral

representation of the mappings by means of the Cauchy integral formula [11]. In particular, in [11], the goal is both the evaluation of the mappings $\phi(h\lambda)$ at the scalar level, assuming that a diagonalization of A is available, but also at the operator level, since it also allows the evaluation of the operators $\phi(hA)$ working with the full matrix A . However, despite the good computational results reported in [11], the way of selecting the parameters involved in the quadrature formulas is not very much clear and they depend very strongly on the equation considered and the spatial discretization parameters. The algorithm we propose is derived by using Laplace transformation formulas and is finally based on a suitable contour integral representation of the mappings $\phi(hA)$, too. However, the quadrature formulas we use borrow their parameters from the method in [14] for the numerical inversion of sectorial Laplace transform, where a rigorous analysis of the error is performed together with an optimization process to choose the different parameters involved in the approximation.

In order to apply the quadrature formulas developed in [14], we derive a representation of the operators in (1.4) and (1.5) as the inverses of suitable Laplace transforms $\Phi(z, hA)$ at certain values of the original variable σ . These Laplace transforms have all the form

$$\Phi(z, hA) = R(z)(zI - hA)^{-1}, \quad (1.8)$$

with $R(z)$ a scalar rational mapping of $z \in \mathbb{C}$. Due to (1.6), the mappings $\Phi(z, hA)$ turn out to be *sectorial* in the variable z , i.e., there exist constants $\gamma \in \mathbb{R}$ and $M > 0$, possibly different from the constants in (1.6), such that

$$\begin{aligned} \Phi(z, hA) \text{ is analytic for } z \text{ in the sector } |\arg(z - \gamma)| < \pi - \delta \text{ and there} \\ \|\Phi(z, hA)\| \leq \frac{M}{|z - \gamma|^\nu}, \quad \text{for some } \nu \geq 1. \end{aligned} \quad (1.9)$$

In this way, we reduce the problem of computing $\phi(hA)$ to the inversion of a sectorial Laplace transform $\Phi(z, hA)$ of the form (1.8). We then use the method for the numerical inversion of sectorial Laplace transforms developed in [14] and reviewed in Section 3. The inversion method consists on a quadrature formula to discretize the inversion formula (3.3), so that we finally approximate

$$\phi(hA) \approx \sum_{\ell=-K}^K w_\ell e^{mz_\ell} \Phi(z_\ell, hA) = \sum_{\ell=-K}^K w_\ell e^{mz_\ell} R(z_\ell)(z_\ell I - hA)^{-1}, \quad (1.10)$$

with the quadrature weights w_ℓ and nodes z_ℓ given in (3.6). The convergence results in [14] assure an error estimate in the approximation (1.10) at least like $O(e^{-cK/\ln K})$. Further, by following this approach, our selection of parameters in the implementation of exponential methods depends only on δ in (1.6), being independent of h and M . In the particular case that we want to evaluate ϕ at a scalar λ in the sector $|\arg(\lambda - \gamma)| \geq \pi - \delta$, the approximation becomes simply

$$\phi(\lambda) \approx \sum_{\ell=-K}^K w_\ell e^{mz_\ell} \Phi(z_\ell, \lambda) = \sum_{\ell=-K}^K w_\ell e^{mz_\ell} \frac{R(z_\ell)}{z_\ell - \lambda} \quad (1.11)$$

and the convergence rate improves to an $O(e^{-cK})$.

The approximation in (1.10) allows the computation of all the operators $\phi(hA)$ required by an exponential method before the time-stepping begins, so that only the

products matrix×vector need to be implemented at every time step. However, formula (1.10) requires the computation and storage of all the inverses $(z_\ell I - hA)^{-1}$, $\ell = -K, \dots, K$. Even if A is a sparse matrix and these inversions can be performed efficiently, the storage of the resulting full matrices $\phi(hA)$ and the subsequent products matrix×vector can become prohibitive for large problems. In this situation, (1.10) should be combined with a data sparse procedure to approximate the resolvent operators, as it is proposed in [4, 5].

Another way to avoid the computation and storage of all the resolvents in (1.10) could be the application of the formula to evaluate the action $\phi(hA)v$ on a given vector v , instead of the full operator $\phi(hA)$. In this case, it is not necessary the computation of the full inverses $(z_\ell I - hA)^{-1}$, but the resolution of the linear systems

$$(z_\ell I - hA)x = v.$$

However, solving all these linear systems for all the quadrature nodes z_ℓ in (1.10) at every time step can become quite expensive and, in this sense, the Krylov approach developed in [7] appears as a more competitive alternative. In this situation, only formula (1.11) can be useful, in order to evaluate the mappings $\phi(h\lambda)$ at the eigenvalues of the Hessenberg matrices involved in the Krylov approximation.

By using (1.10) we are in fact computing an approximation to the numerical solution of (1.1) provided by an exponential method. Thus, the global error after applying (1.10) to the time integration of (1.1) can be split into the error in the time integration by the “pure” exponential method and the deviation from the numerical solution introduced by the approximation (1.10) of the operators $\phi(hA)$. The error in the time integration for the exponential integrators considered in the present paper is analyzed in [2] and [10] (see Section 2), and the quadrature error is analyzed in [14] (see Section 3, Theorem 3.1). In order to visualize the effect of this approximation, we show in Section 6 the performance of our implementation for several problems with known exact solution and moderate size after the spatial discretization. In the error plots provided we can observe that the error coincides with the expected error for the exact exponential integrators up to high accuracy for quite moderate values of K in (1.10), i.e., the error induced by the quadrature (1.10) is negligible compared to the error in the time integration.

Finally we notice that the matrix exponential e^{tA} and also certain rational approximations to it originating from Runge–Kutta schemes have already been successfully approximated by using this approach [12, 13, 14].

The paper is organized as follows. In Section 2 we consider the class of explicit multistep exponential methods proposed in [2] and the explicit exponential Runge–Kutta methods in [10]. Section 3 is a review of the method for the numerical inversion of sectorial Laplace transforms presented in [14]. In Section 4 we deduce a representation for the operators required in the implementation of these integrators in terms of suitable Laplace transforms and apply the inversion method to the implementation of exponential methods. In Section 5 we consider with some detail the evaluation of the associated mappings at the scalar level and present some numerical results. We finally test our algorithm with several examples in Section 6, where we implemented (1.10) by using the full matrices.

2. Exponential methods. In this section we review some of the exponential methods in the recent literature. In this way, we consider the class of explicit multistep exponential integrators developed in [2] and the exponential Runge–Kutta methods of [10], which are explicit as well.

2.1. Multistep exponential methods. In [2], a class of explicit exponential methods is constructed for the time integration of problems of the form (1.1) with $A : D(A) \subset X \rightarrow X$ the infinitesimal generator of a C_0 -semigroup e^{tA} , $t \geq 0$, of linear and bounded operators in a Banach space X .

As we already stated in the Introduction, we will restrict ourselves to the case of A in (1.1) sectorial. Then, for $0 \leq \alpha < 1$, the fractional powers $(\omega - A)^\alpha$ are well defined for $\omega > \gamma$ in (1.6), and $X_\alpha = D((\omega - A)^\alpha)$ endowed with the graph norm $\|\cdot\|_\alpha$ is a Banach space [6]. In this situation, the nonlinearity f in (1.1) is assumed to be defined on $[0, T] \times X_\alpha \rightarrow X$, for some $0 \leq \alpha < 1$, and to be locally Lipschitz in a strip along the exact solution. Thus, there exists $L(R, T) > 0$ such that

$$\|f(t, \eta) - f(t, \xi)\| \leq L\|\eta - \xi\|_\alpha, \quad \eta, \xi \in X_\alpha, \quad 0 \leq t \leq T, \quad (2.1)$$

for $\max(\|\eta - u(t)\|_\alpha, \|\xi - u(t)\|_\alpha) \leq R$.

For $k \geq 1$, the k -step method is constructed from the variation of constants formula in the interval $[t_n, t_{n+k}]$ as follows. Taking a stepsize $h = T/N$, $N \geq k$, and the corresponding time levels $t_n = nh$, $0 \leq n \leq N$, the solution u of (1.1) at t_{n+k} is given by

$$u(t_{n+k}) = e^{khA}u(t_n) + h \int_0^k e^{(k-\sigma)hA} f(t_n + \sigma h, u(t_n + \sigma h)) d\sigma. \quad (2.2)$$

Given approximations $u_{n+j} \approx u(t_{n+j})$, $0 \leq j \leq k-1$, the approximation $u_{n+k} \approx u(t_{n+k})$ is obtained after replacing f in (2.2) by the Lagrange interpolation polynomial of degree $k-1$, $P_{n,k-1}$ through the points $\{(t_{n+j}, f(t_{n+j}, u_{n+j}))\}_{j=0}^{k-1}$ and integrating exactly. Writing

$$P_{n,k-1}(t_n + \sigma h) = \sum_{j=0}^{k-1} \binom{\sigma}{j} \Delta^j f_n, \quad (2.3)$$

with $f_m = f(t_m, u_m)$, $0 \leq m \leq N-1$, and Δ the standard forward difference operator, the approximation u_{n+k} is computed from

$$u_{n+k} = \phi_0(k, hA)u_n + h \sum_{j=0}^{k-1} \phi_{j+1}(k, hA) \Delta^j f_n, \quad (2.4)$$

where, for $\lambda \in \mathbb{C}$, $k \geq 1$, and $0 \leq j \leq k$, the mappings $\phi_j(k, \lambda)$ are given in (1.3). The methods defined in (2.4) are explicit and, as we already mentioned in the Introduction, they require the evaluation of $\phi_j(k, hA)$.

In [2, Theorem 1] it is shown that if the nonlinearity $f(t, u(t))$ belongs to the space $C^k([0, T], X)$ and the starting values $u_0, \dots, u_{k-1} \in X_\alpha$ satisfy

$$\|u(t_j) - u_j\|_\alpha \leq C_0 h^k, \quad 0 \leq j \leq k-1,$$

the method defined in (2.4) exhibits full order k , i.e., there exists $C > 0$ such that

$$\|u(t_n) - u_n\|_\alpha \leq C h^k \|f^{(k)}\|_\infty, \quad 0 \leq n \leq N. \quad (2.5)$$

The constant C depends on k, α, T , $L(R, T)$ in (2.1), and γ, M in (1.6), but it is independent of h and f .

2.2. Exponential Runge–Kutta methods. Explicit exponential Runge–Kutta methods are presented in [10] for the time integration of semilinear parabolic problems. For $h = T/N$, $N \geq 1$, and $1 \leq i \leq s$, the approximations u_n to $u(t_n)$, with $t_n = nh$, are given by

$$\begin{aligned} U_{ni} &= e^{c_i h A} u_n + h \sum_{j=1}^{i-1} a_{ij}(hA) f(t_n + c_j h, U_{nj}), \\ u_{n+1} &= e^{hA} u_n + h \sum_{i=1}^s b_i(hA) f(t_n + c_i h, U_{ni}), \end{aligned} \quad (2.6)$$

with $c_1 = 0$ ($U_{n1} = u_n$). The coefficients $b_i(\lambda)$ and $a_{ij}(\lambda)$ are linear combinations of $\varphi_k(\lambda)$ and $\varphi_k(c_l \lambda)$ with

$$\varphi_k(\lambda) = \int_0^1 e^{(1-\sigma)\lambda} \frac{\sigma^{k-1}}{(k-1)!} d\sigma, \quad \lambda \in \mathbb{C}, \quad k \geq 1, \quad t > 0. \quad (2.7)$$

Setting $\varphi_0(\lambda) = e^\lambda$, we see that the implementation of (2.6) requires the evaluation of $\varphi_k(hA)$ and $\varphi_k(c_l hA)$, for $1 \leq l \leq s$ and several values of $k \geq 0$.

As in [2], the nonlinearity f in (1.1) is assumed to satisfy a local Lipschitz condition like (2.1) and the solution $u : [0, T] \rightarrow X_\alpha$ and $f : [0, T] \times X_\alpha \rightarrow X$ are assumed to be sufficiently smooth so that the composition

$$f^* : [0, T] \times X_\alpha \rightarrow X : t \rightarrow f^*(t) = f(t, u(t))$$

is a sufficiently often differentiable mapping, too. Under these assumptions, stiff order conditions are derived and exponential Runge–Kutta methods of the form (2.6) are constructed up to order four in [10].

3. The numerical inversion of sectorial Laplace transforms. In this section we review the numerical inversion method for sectorial Laplace transforms presented in [14], which is a further development of [13].

For a locally integrable mapping $f : (0, \infty) \rightarrow X$, bounded by

$$\|f(t)\| \leq C t^{\nu-1} e^{\gamma t}, \quad \text{for some } \gamma \in \mathbb{R}, \quad \nu > 0, \quad (3.1)$$

we denote its Laplace transform

$$F(z) = \mathcal{L}[f](z) = \int_0^\infty e^{-tz} f(t) dt, \quad \operatorname{Re} z > \gamma. \quad (3.2)$$

When F satisfies (1.9), the method in [14] allows to approximate the values of f from few evaluations of F . This is achieved by means of a suitable quadrature rule to discretize the inversion formula

$$f(t) = \frac{1}{2\pi i} \int_\Gamma e^{zt} F(z) dz, \quad (3.3)$$

where Γ is a contour in the complex plane, running from $-i\infty$ to $i\infty$ and laying in the analyticity region of F . Due to (1.9), Γ can be taken so that it begins and ends in the half plane $\operatorname{Re} z < 0$. Following [14], in (3.3) we choose Γ as the left branch of a hyperbola parameterized by

$$\mathbb{R} \rightarrow \Gamma : x \mapsto T(x) = \mu(1 - \sin(\alpha + ix)) + \gamma, \quad (3.4)$$

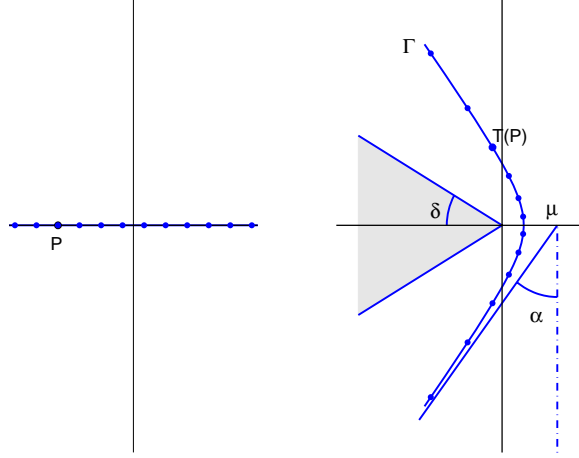


FIG. 3.1. Action of T in (3.4) on the real axis

where $\mu > 0$ is a scale parameter, γ is the shift in (1.9), and $0 < \alpha < \frac{\pi}{2} - \delta$. Thus, Γ is the left branch of the hyperbola with center at $(\mu, 0)$, foci at $(0, 0)$, $(2\mu, 0)$, and with asymptotes forming angles $\pm(\pi/2 + \alpha)$ with the real axis, so that Γ remains in the sector of analyticity of F , $|\arg(z - \gamma)| < \pi - \delta$. In Figure 3.1 we show the action of the conformal mapping T on the real axis.

After parameterizing (3.3), the function f is approximated by applying the truncated trapezoidal rule to the resulting integral along the real axis, i.e.,

$$f(t) = \frac{1}{2\pi i} \int_{\Gamma} e^{tz} F(z) dz \approx \sum_{\ell=-K}^K w_{\ell} e^{tz_{\ell}} F(z_{\ell}), \quad (3.5)$$

with quadrature weights w_{ℓ} and nodes z_{ℓ} given by

$$w_{\ell} = -\frac{\tau}{2\pi i} T'(\ell\tau), \quad z_{\ell} = T(\ell\tau), \quad -K \leq \ell \leq K, \quad (3.6)$$

and $\tau > 0$ a suitable step length parameter. We notice that the minus sign in the formula for the weights comes from setting the proper orientation in the parametrization of Γ . In case of symmetry, the sum in (3.5) can be halved to

$$f(t) \approx \operatorname{Re} \left(\sum_{\ell=0}^K w_{\ell}^* e^{tz_{\ell}} F(z_{\ell}) \right), \quad (3.7)$$

with $w_0^* = w_0$ and $w_{\ell}^* = 2w_{\ell}$, $\ell \geq 1$. The good behavior of the quadrature formula (3.5) is due to the good properties of the trapezoidal rule when the integrand can be analytically extended to a horizontal strip around the real axis [17, 18].

During the last years, different choices of contours Γ and parameterizations have been studied for the numerical inversion of sectorial Laplace transforms. Apart from the approach in [13, 14], which is the one we follow, the choice of a hyperbola has been studied in [4, 5, 15, 16, 21]. The choice of Γ as a parabola has been considered recently in [4, 5, 21] and we refer also to Talbot's method [19, 20] for another kind of integration contour Γ , with horizontal asymptotes as $|z| \rightarrow \infty$.

The error analysis for (3.5) shows that the role played by the round-off errors is very important. Several ways to avoid error propagation are proposed in [14], depending on the available information about the errors in the evaluations of $F(z_\ell)$ and the elementary mappings involved in (3.5). Our algorithm for exponential methods uses (1.10) to approximate the required operators $\phi(hA)$ in (1.4) and (1.5), i.e., (3.5) with Laplace transforms of the form (1.8). Thus, the required Laplace transforms will be in practice evaluated by performing the inversion of few matrices of the form $zI - hA$, with A being normally a discrete version of the operator in (1.1) and h the time step length. Taking into account that this is likely to be accomplished by means of some auxiliary routine, we propose to use in this context the least accurate version of the method in [14], where the error propagation is avoided without using any information about the errors in the evaluations of (3.5).

The following result provides an error bound for the proposed version of the inversion method like $O(e^{-cK/\ln K})$, with K the number of quadrature nodes.

THEOREM 3.1. [14] *Assume that the Laplace transform $F(z)$ satisfies the sectorial condition (1.9) and let α and d be such that*

$$0 < \alpha - d < \alpha + d < \frac{\pi}{2} - \delta. \quad (3.8)$$

For $t_0 > 0$, $\Lambda \geq 1$ and $K \geq 1$, we select the parameters

$$\tau = \frac{a(K)}{K}, \quad \mu = \frac{2\pi d}{\Lambda t_0 a(K)}, \quad (3.9)$$

with $a(K) = \operatorname{arccosh}(\Lambda K / \sin \alpha)$.

Then, the error $E_K(t)$ in the approximation (3.5) to $f(t)$ with quadrature weights and nodes in (3.6) is bounded by

$$\|E_K(t)\| \leq M \Pi Q e^{2\pi d/a(K)} t^{\nu-1} \left(\varepsilon + \frac{e^{-2\pi dK/a(K)}}{1 - e^{-2\pi dK/a(K)}} \right), \quad (3.10)$$

uniformly for $t \in [t_0, \Lambda t_0]$, where M and ν are the constants in (1.9), ε is the precision in the evaluations of the Laplace transform F and the elementary operations in (3.5),

$$\Pi = \frac{1}{\pi} \sqrt{\frac{1 + \sin(\alpha + d)}{(1 - \sin(\alpha + d))^{2\nu-1}}}$$

and

$$Q = \max\{4L(\lambda t_0 \sin(\alpha - d)), \tau + L(\lambda t_0 \sin \alpha)\},$$

with $L(x) = 1 - \ln(1 - e^{-x})$.

In case we have some reliable information about the errors in the computation of the matrices $(zI - hA)^{-1}$, an ε -depending selection of τ and λ improves the error bound (3.10) to $O(e^{-cK})$, i.e., a pure exponentially decaying bound in K . In this situation, given ε , K , and α, d fulfilling (3.8), the parameters τ and μ are given by

$$\tau = \frac{a(\theta^*)}{K}, \quad \mu = \frac{2\pi dK(1 - \theta^*)}{\Lambda t_0 a(\theta^*)}, \quad (3.11)$$

where, for $\theta \in (0, 1)$, $a(\theta)$ is the mapping

$$a(\theta) = \operatorname{arccosh} \left(\frac{\Lambda}{(1 - \theta) \sin \alpha} \right), \quad (3.12)$$

and

$$\theta^* = \min_{\theta \in (0,1)} \left(\varepsilon e^{2\pi dK(1-\theta)/a(\theta)} + e^{-2\pi dK\theta/a(\theta)} \right). \quad (3.13)$$

Given $K \geq 1$, the expression to be minimize in (3.13) represents the main part of the error bound obtained in [14] for any fixed $\theta \in (0,1)$. By choosing for every K the optimal value θ^* in (3.13), an error bound like $O(e^{-cK})$ is attained. We notice that the error bound stated in Theorem 3.1 is obtained for $\theta^* = 1 - 1/K$ in (3.11), instead of θ^* in (3.13). We refer to [14] for details. There, the case $\nu < 1$ in (1.9) is also studied.

4. Evaluation of the vector-valued mappings. In this section we apply some Laplace transformation formulas to obtain a suitable representation of the operators $\phi_j(k, hA)$, $\varphi_j(hA)$, and $\varphi_j(c_l hA)$ required in (2.4) and (2.6).

Let us denote the Laplace transform of a mapping $f(\sigma)$ by $F(z) = \mathcal{L}[f](z)$, and the inverse Laplace transform by $f(\sigma) = \mathcal{L}^{-1}[F](\sigma)$.

4.1. Evaluation of the mappings required by the multistep methods.

For ϕ_j in (1.3) with $1 \leq j \leq k$, it holds

$$\phi_j(k, \lambda) = \int_0^k e^{(k-\sigma)\lambda} \begin{pmatrix} \sigma \\ j \end{pmatrix} d\sigma = \mathcal{L}^{-1}[\mathcal{L}[f_0(\cdot, \lambda)] \times \mathcal{L}[f_j]](k),$$

where, for $\sigma > 0$,

$$f_0(\sigma, \lambda) = e^{\sigma\lambda} \quad \text{and} \quad f_j(\sigma) = \begin{pmatrix} \sigma \\ j \end{pmatrix}. \quad (4.1)$$

For every $j \geq 1$ and $z \in \mathbb{C}$, we define

$$\Phi_j(z, \lambda) = \mathcal{L}[f_0(\cdot, \lambda)](z) \times \mathcal{L}[f_j](z) = \frac{1}{z - \lambda} \times \mathcal{L}[f_j](z). \quad (4.2)$$

Then, for every $\lambda \in \mathbb{C}$ and $j \geq 1$,

$$\phi_j(k, \lambda) = \mathcal{L}^{-1}[\Phi_j(\cdot, \lambda)](k). \quad (4.3)$$

For $j = 0$

$$\phi_0(k, \lambda) = e^{k\lambda} = \mathcal{L}^{-1} \left(\frac{1}{\cdot - \lambda} \right) (k),$$

and thus we define

$$\Phi_0(z, \lambda) = \frac{1}{z - \lambda}. \quad (4.4)$$

For λ scalar, the mappings $\Phi_j(z, \lambda)$, with $1 \leq j \leq 4$ are given by

$$\begin{aligned} \Phi_1(z, \lambda) &= \frac{1}{z(z - \lambda)}, & \Phi_2(z, \lambda) &= \frac{1}{z^2(z - \lambda)}, \\ \Phi_3(z, \lambda) &= \frac{2 - z}{2z^3(z - \lambda)}, & \Phi_4(z, \lambda) &= \frac{3 - 3z + z^2}{3z^4(z - \lambda)}. \end{aligned} \quad (4.5)$$

In order to evaluate $\phi_j(k, hA)$, $0 \leq j \leq 4$, we propose to use the formulas in (4.5) with hA instead of λ and perform the inversion of the Laplace transform to approximate the original mappings at $\sigma = k$. In this way, the Laplace transforms we need to invert are:

$$\begin{aligned}
\Phi_0(z, hA) &= (zI - hA)^{-1}, \\
\Phi_1(z, hA) &= \frac{1}{z}(zI - hA)^{-1}, \\
\Phi_2(z, hA) &= \frac{1}{z^2}(zI - hA)^{-1}, \\
\Phi_3(z, hA) &= \frac{2-z}{2z^3}(zI - hA)^{-1}, \\
\Phi_4(z, hA) &= \frac{3-3z+z^2}{3z^4}(zI - hA)^{-1}.
\end{aligned} \tag{4.6}$$

Although the formulas in (4.6) are derived just formally, we notice that they can be justified by combining the Cauchy integral formula with the inversion formula for the Laplace transform. More precisely, for suitable contours Γ_1 and Γ_2 in the complex plane, both laying in the resolvent set of A , it holds

$$\begin{aligned}
\phi_j(k, hA) &= \frac{1}{2\pi i} \int_{\Gamma_1} \phi_j(k, \lambda)(\lambda I - hA)^{-1} d\lambda \\
&= \frac{1}{2\pi i} \int_{\Gamma_1} \left(\frac{1}{2\pi i} \int_{\Gamma_2} e^{k\xi} \Phi_j(\xi, \lambda) d\xi \right) (\lambda I - hA)^{-1} d\lambda \\
&= \frac{1}{2\pi i} \int_{\Gamma_2} e^{k\xi} \left(\frac{1}{2\pi i} \int_{\Gamma_1} \Phi_j(\xi, \lambda)(\lambda I - hA)^{-1} d\lambda \right) d\xi \\
&= \frac{1}{2\pi i} \int_{\Gamma_2} e^{k\xi} \Phi_j(\xi, hA) d\xi = \mathcal{L}^{-1}[\Phi_j(\cdot, hA)](k).
\end{aligned} \tag{4.7}$$

Due to (1.6), all the Laplace transforms $\Phi_j(k, hA)$ in (4.6) are sectorial, since they satisfy (1.9) with $\gamma^* = \max\{0, \gamma\}$ and $\delta^* = \delta$, for γ and δ in (1.6). We notice that, for all j , the resulting bounds in (1.9) are independent of h .

Thus, we can compute the operators $\phi_j(k, hA)$, $0 \leq j \leq k$, by using the method described in Section 3 to compute the inverse Laplace transforms of the mappings $\Phi_j(z, hA)$ in (4.2). We notice that the inverse Laplace transforms need to be approximated only at the fix value $\sigma = k$, which is specially favorable for the application of the inversion method (see the bound in Theorem 3.1). Then, we set $\Lambda = 1$, $t_0 = k$, and select the parameters τ and μ following Theorem 3.1. The selection of α and d is more heuristic and a good choice is $\alpha \approx \frac{1}{2}(\frac{\pi}{2} - \delta)$ and d slightly smaller than α . For example, if $\delta = 0$ in (1.9), good values are around $\alpha = 0.7$ and $d = 0.6$. Next, we compute the quadrature weights w_ℓ and nodes z_ℓ in (3.6) and approximate the operators in (2.4) by

$$\phi_j(k, hA) \approx \sum_{\ell=-K}^K w_\ell e^{kz_\ell} \Phi_j(z_\ell, hA). \tag{4.8}$$

The sum in (4.8) can be halved in case of symmetry like in (3.7).

As we already mentioned in the Introduction, the computation of all the required operators $\phi_j(k, hA)$, $0 \leq j \leq k$, can be carried out before the time stepping of

the exponential method begins. Thus, if we use the method of lines and apply the exponential method to some spatial discretization of (1.1), only the matrix-vector products in (2.4) need to be computed at every time step.

4.2. Evaluation of the mappings required by the Runge–Kutta methods. For φ_j in (2.7), $j \geq 1$ and $t > 0$, we have

$$\varphi_j(\lambda) = \int_0^1 e^{(1-\sigma)\lambda} \frac{\sigma^{j-1}}{(j-1)!} d\sigma = \mathcal{L}^{-1}[\mathcal{L}[g_0(\cdot, \lambda)] \times \mathcal{L}[g_j]](1),$$

where, for $\sigma > 0$ and $\lambda \in \mathbb{C}$,

$$g_0(\sigma, \lambda) = e^{\sigma\lambda} \quad \text{and} \quad g_j(\sigma) = \frac{\sigma^{j-1}}{(j-1)!}. \quad (4.9)$$

For every $j \geq 1$ and $z \in \mathbb{C}$, we define

$$\Psi_j(z, \lambda) = \mathcal{L}[g_0(\cdot, \lambda)](z) \times \mathcal{L}[g_j](z) = \frac{1}{z^j(z-\lambda)} \quad (4.10)$$

and $\Psi_0(z, \lambda) = (z-\lambda)^{-1}$. Then, for every $\lambda \in \mathbb{C}$ and $j \geq 0$,

$$\varphi_j(\lambda) = \mathcal{L}^{-1}[\Psi_j(\cdot, \lambda)](1). \quad (4.11)$$

The same argument as in (4.7) justifies the computation of the operators $\varphi_j(hA)$ and $\varphi_j(c_l hA)$, $j \geq 0$, $2 \leq l \leq s$, by performing the inversion of the Laplace transforms

$$\Psi_j(z, \beta hA) = \frac{1}{z^j} (zI - \beta hA)^{-1}, \quad j \geq 0, \quad \beta = 1, c_l, \quad (4.12)$$

to approximate the original mappings at $\sigma = 1$. If (1.6) holds, the Laplace transforms in (4.12) are also the sectorial in the sense of (1.9) and we can use the inversion method of [14].

As in the case of the methods in (2.4), the computation of all the required operators $\varphi_j(hA)$ and $\varphi_j(c_l hA)$, $j \geq 0$, can be carried out before the time stepping of the exponential method begins.

REMARK 1. *In general, we can always evaluate a mapping $\phi(hA)$ of the form of (1.5) by using the numerical inversion of the Laplace transform, just by noticing that $\phi(hA)$ is the inverse Laplace transform at $\sigma = n$ of a mapping $\Phi(z, hA)$ like in (1.8),*

$$\Phi(z, hA) = R(z)(zI - hA)^{-1},$$

with $R(z) = \mathcal{L}[p](z)$, a scalar rational function of z .

The above Remark implies that our algorithm can be used to implement other kinds of exponential methods, different than those in [2, 10], as long as they require the evaluation of mappings of the form of (1.4) and (1.5).

5. Evaluation of the scalar mappings. As we already mentioned in the Introduction, we can also apply the inversion of the Laplace transform to evaluate with accuracy the scalar mappings $\phi(\lambda)$ in (1.5). In this section we consider with some detail the evaluation of the mappings

$$g_j(m, \lambda) = \int_0^m e^{(m-\sigma)\lambda} \sigma^{j-1} d\sigma, \quad j \geq 1, \quad m \in \mathbb{N}, \quad (5.1)$$

by means of the quadrature formula (1.11). The result provided by (1.11) does not depend on the size of λ , but the formula is only useful in principle for values of λ inside a sector of the form $|\arg(\lambda - \gamma)| \geq \pi - \delta$. However, using that

$$e^{m\lambda} g_1(m, -\lambda) = g_1(m, \lambda), \quad \lambda \in \mathbb{C}, \quad (5.2)$$

and

$$g_{j+1}(m, \lambda) = \frac{j g_j(m, \lambda) - m^j}{\lambda}, \quad j \geq 1, \quad m \in \mathbb{N}, \quad (5.3)$$

it is easy to see by induction that, for $m \in \mathbb{N}$ and $\lambda \in \mathbb{C}$,

$$e^{m\lambda} g_j(m, -\lambda) = \sum_{\ell=1}^j \binom{j-1}{\ell-1} (-1)^{\ell-1} m^{j-\ell} g_\ell(m, \lambda), \quad j \geq 1. \quad (5.4)$$

Thus, we can compute

$$g_j(m, -\lambda) = e^{-m\lambda} \mathcal{L}^{-1} [G_j^*(\cdot, \lambda)](m), \quad (5.5)$$

with

$$G_j^*(z, \lambda) = \frac{1}{z^j(z-\lambda)} \sum_{\ell=1}^j \binom{j-1}{\ell-1} \ell! (-1)^{\ell-1} (mz)^{j-\ell}, \quad j \geq 1. \quad (5.6)$$

which provides a stable formula to approximate $g_j(m, -\lambda)$ for λ inside a proper sector $|\arg(\lambda - \gamma)| > \pi - \delta$ and moderate size.

In Table 5.1 we show the error obtained in the evaluation of $\varphi(\lambda) = g_1(1, \lambda)$ in (1.7) for different values of λ in the interval $[-1, 1]$. For $\lambda < 0$, we applied the inversion formula (3.5) with $t = 1$ and

$$F(z) = G_1(z, \lambda) = \frac{1}{z(z-\lambda)}, \quad (5.7)$$

which, for these values of λ , fulfils (1.9) with $\delta = 0$, $\gamma = 0$, and $\nu = 2$. We assumed that the evaluations of G_1 can be carried out in MATLAB up to machine accuracy and thus we set $\varepsilon = 2.2204 \times 10^{-16}$. Then, we computed the quadrature weights and nodes in (3.6) following (3.11)–(3.13) with $\Lambda = 1$. Setting $\alpha = 0.7$ and $d = 0.6$, we obtained $\theta = 0.693$, for $K = 15$, and $\theta = 0.793$, for $K = 25$. In Table 5.1 we can see that $K = 25$ is enough to attain almost the machine accuracy of MATLAB in the evaluations of $\varphi(\lambda)$. For positive values of λ , we used (5.2) with $m = 1$.

6. Numerical illustrations. In this section we test our algorithm by considering the same examples as in [2] and [10].

6.1. Example for the multistep exponential methods. Our first example is the problem considered in [2]

$$u_t(x, t) = u_{xx}(x, t) + \left(\int_0^1 u(s, t) ds \right) u_x(x, t) + g(x, t), \quad (6.1)$$

for $x \in [0, 1]$ and $t \in [0, 1]$, subject to homogeneous Dirichlet boundary conditions and with $g(x, t)$ such that the exact solution to (6.1) is $u(x, t) = x(1-x)e^t$

TABLE 5.1

Computation of $\varphi(\lambda)$ in (1.7) for $\lambda \in [-1, 1]$ by using formulas (3.5) and (5.2). We show the absolute error obtained in MATLAB with $K = 15$ and $K = 25$.

$\lambda < 0$	$K = 15$	$K = 25$	$-\lambda$	$K = 15$	$K = 25$
-1	1.5050e-12	1.3323e-15	1	1.5050e-12	3.3307e-15
-1e-1	1.5227e-12	3.2196e-15	1e-1	1.5227e-12	3.5527e-15
-1e-2	1.4243e-12	4.4409e-15	1e-2	1.4243e-12	4.6629e-15
-1e-3	1.3750e-12	1.3323e-15	1e-3	1.3750e-12	1.3323e-15
-1e-4	1.3738e-12	1.7764e-15	1e-4	1.3738e-12	1.7764e-15
-1e-5	1.3747e-12	3.6637e-15	1e-5	1.3747e-12	3.7748e-15
-1e-6	1.3748e-12	3.6637e-15	1e-6	1.3748e-12	3.7748e-15
-1e-7	1.3695e-12	1.9984e-15	1e-7	1.3695e-12	1.9984e-15
-1e-8	1.3717e-12	1.1102e-16	1e-8	1.3717e-12	2.2204e-16
-1e-9	1.3715e-12	1.1102e-16	1e-9	1.3715e-12	0
-1e-10	1.3711e-12	0	1e-10	1.3711e-12	0
-1e-11	1.3711e-12	0	1e-11	1.3711e-12	0
-1e-12	1.3715e-12	1.1102e-16	1e-12	1.3715e-12	0
-1e-13	1.3712e-12	0	1e-13	1.3712e-12	2.2204e-16

The spatial discretization of (6.1) is carried out by using standard finite differences with $J = 512$ spatial nodes, centered for the approximation of u_x . The nonlocal term is approximated by means of the composite Simpson's formula.

To integrate in time the semidiscrete problem we use (2.4) with $k = 1, 2, 3$ and 4, so that A is the $(J - 1) \times (J - 1)$ matrix

$$A = J^2 \text{tridiag}([1, -2, 1]).$$

We approximate the matrices $\phi_j(k, hA)$, $0 \leq j \leq k$, required in (2.4) by applying the quadrature rule (4.8). To avoid an extra source of error, the initial values u_1, \dots, u_{k-1} are computed from the exact solution. In a less academic example, these values can be computed by means of a one-step method of sufficiently high order or by performing the fix point iteration proposed in [2].

In Figure 6.1 we show the error versus the stepsize at $t = 1$, measured in a discrete version of the norm $\|\cdot\|_{1/2}$, for $K = 25$ and $K = 35$ in (4.8). We see that for $K = 35$ the full precision is achieved for all the methods implemented; cf. [2, Section 6]. In Figure 6.1 we also show lines of slope 1, 2, 3 and 4, to visualize the order of convergence. In fact, we can observe that the order of convergence for this example is slightly higher than the one predicted in [2], approximately 2.15, 3.15 and 4.15 for $k = 2, 3$ and 4, respectively, instead of sharp order k . A further study of this behavior is beyond the scope of this paper.

6.2. Examples for the exponential Runge–Kutta methods. For the following two examples we consider the problems and some of the integrators proposed in [10]. Following the notation in [10], in the Butcher tableaus below we use the abbreviations

$$\varphi_i = \varphi_i(hA), \quad \text{and} \quad \varphi_{i,j} = \varphi_{i,j}(hA) = \varphi_i(c_j hA), \quad 2 \leq j \leq s. \quad (6.2)$$

Thus, our second example is

$$u_t(x, t) = u_{xx}(x, t) + \frac{1}{1 + u(x, t)^2} + g(x, t), \quad (6.3)$$

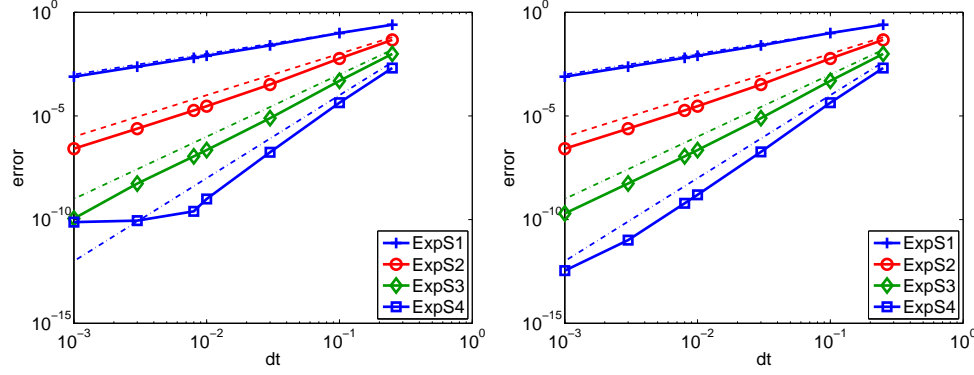


FIG. 6.1. Error of exponential multistep methods (2.4) applied to (6.1), for $k = 1, 2, 3$, and 4. Left: With $K = 25$ quadrature nodes on the hyperbolas, Right: With $K = 35$.

for $x \in [0, 1]$ and $t \in [0, 1]$, subject to homogeneous Dirichlet boundary conditions and with $g(x, t)$ such that the exact solution to (6.3) is again $u(x, t) = x(1 - x)e^t$.

We discretize this problem in space by standard finite differences with $J = 200$ grid points. For the time integration of the semidiscrete problem, we implemented (2.6) with $s = 1$, the second-order method

$$\begin{array}{c|c} 0 & \\ \hline \frac{1}{2} & \frac{1}{2}\varphi_{1,2} \\ \hline & 0 \quad \varphi_1 \end{array} \quad (6.4)$$

the third-order method

$$\begin{array}{c|ccc} 0 & & & \\ \hline \frac{1}{3} & \frac{1}{3}\varphi_{1,2} & & \\ \frac{2}{3} & \frac{2}{3}\varphi_{1,3} - \frac{4}{3}\varphi_{2,3} & \frac{4}{3}\varphi_{2,3} & \\ \hline & \varphi_1 - \frac{3}{2}\varphi_2 & 0 & \frac{3}{2}\varphi_2 \end{array} \quad (6.5)$$

and the fourth-order one

$$\begin{array}{c|cccccc} 0 & & & & & \\ \hline \frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & \\ \frac{1}{2} & \frac{1}{2}\varphi_{1,3} - \varphi_{2,3} & \varphi_{2,3} & & & \\ 1 & \varphi_{1,4} - 2\varphi_{2,4} & \varphi_{2,4} & \varphi_{2,4} & & \\ \hline \frac{1}{2} & \frac{1}{2}\varphi_{1,5} - 2a_{5,2} - a_{5,4} & a_{5,2} & a_{5,2} & a_{5,4} & \\ \hline & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 0 & 0 & -\varphi_2 + 4\varphi_3 & 4\varphi_2 - 8\varphi_3 \end{array} \quad (6.6)$$

with

$$a_{5,2} = \frac{1}{2}\varphi_{2,5} - \varphi_{3,4} + \frac{1}{4}\varphi_{2,4} - \frac{1}{2}\varphi_{3,5}$$

and

$$a_{5,4} = \frac{1}{4}\varphi_{2,5} - a_{5,2}.$$

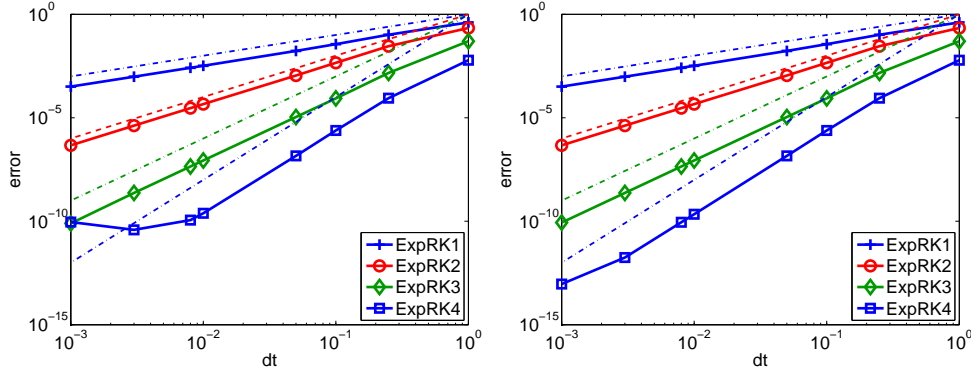


FIG. 6.2. Error of Runge-Kutta methods (2.4) with $s = 1$, (6.4), (6.5), and (6.6), applied to (6.3). Left: With $K = 25$ quadrature nodes in (4.8), Right: With $K = 35$.

For the implementation of (6.4) we need to invert four different Laplace transforms of the form of (4.12), to approximate $\varphi_0(\frac{h}{2}A)$, $\varphi_0(hA)$, $\varphi_1(\frac{h}{2}A)$, and $\varphi_1(hA)$. The implementation of both (6.5) and (6.6) requires the inversion of eight Laplace transforms.

In Figure 6.2 we show the error at $t = 1$ versus the stepsize, measured in the maximum norm. The expected order of convergence for this example is k for the k -order method. In order to check our algorithm, we added lines with the corresponding slopes in Figure 6.2. We can see that also for this kind of methods we attain full precision for $K = 35$ in (4.8).

Finally we consider the second example in [10]

$$u_t(x, t) = u_{xx}(x, t) + \int_0^1 u(s, t) ds + g(x, t), \quad (6.7)$$

for $x \in [0, 1]$ and $t \in [0, 1]$, subject also to homogeneous Dirichlet boundary conditions and with $g(x, t)$ such that the exact solution to (6.7) is $u(x, t) = x(1 - x)e^t$.

We discretize this problem in space as in the previous example (6.3), and use the composite Simpson's rule for the approximation of the nonlocal term. For the time integration, we use (2.6) with $s = 1$, (6.4), and (6.5). In Figure 6.3 we show the error at $t = 1$, measured in a discrete version of the norm $\|\cdot\|_{1/2}$; cf. [10, Section 6]. The expected order of convergence for this example is 1 for the first order method, 1.75 for (6.4), and 2.75 for (6.5). In order to test our algorithm for the exponential methods, we added lines with the corresponding slopes in Figure 6.3. We refer to [10] for a detailed explanation of the order reduction phenomenon in this example. The less stringent accuracy requirements in this case allow us to achieve full precision with only $K = 25$ quadrature nodes in (4.8).

7. Conclusions. In this paper we derived a way to approximate the exponential-like operators required for the implementation of different kinds of exponential methods present in the recent literature for the time integration of semilinear problems. The approach is based on the numerical inversion of the Laplace transform and its application is restricted to parabolic problems. When applicable, the proposed algorithm is shown to be very efficient, both at the scalar level and the operator level, and it is quite simple to implement. Apart from the error bound stated in Theorem 3.1,

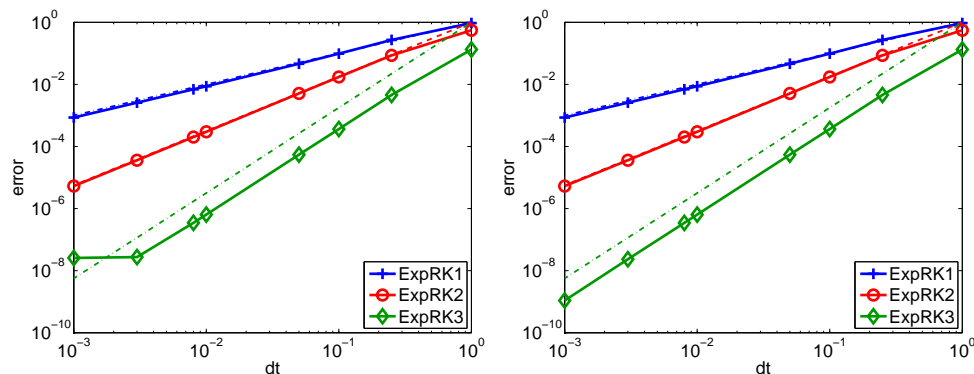


FIG. 6.3. Error of Runge-Kutta methods (2.6) with $s = 1$, (6.4), and (6.5), applied to (6.7). Left: With $K = 20$ quadrature nodes in (4.8), Right: With $K = 25$.

which is a partial result of those proved in [14], we tested the algorithm with several academic examples from [2] and [10] and with the evaluation of the prototypical mapping φ in (1.7).

The comparison of our algorithm with other techniques present in the literature (see the Introduction) is beyond the scope of the present paper. Some testing for comparison of CPU times and storage requirements with problems of different complexity could be the subject of future work.

8. Acknowledgements. The author is grateful to Enrique Zuazua for proposing a good problem and for useful discussions during the preparation of the manuscript. She also thanks César Palencia for suggesting the use of exponential methods and Christian Lubich for his valuable comments on a preliminary version of the paper.

REFERENCES

- [1] H. BERLAND, B. SKAFLESTAD, W. M. WRIGHT, *EXPINT - A MATLAB package for exponential integrators*, ACM Transactions on Mathematical Software, 33 (2007), Article Number 4.
- [2] M. P. CALVO, C. PALENCIA, *A class of explicit multistep exponential integrators for semilinear problems*, Numer. Math. 102 (2006), pp. 367–381.
- [3] S. M. COX, P. C. MATTHEWS, *Exponential time differencing for stiff systems*, J. Comput. Phys. 176 (2002), pp. 430–455.
- [4] I. P. GAVRILYUK, W. HACKBUSCH, B. N. KHOROMSKIY, *Data-sparse approximation to the operator-valued functions of elliptic operators*, Math. Comp., 73 (2004), pp. 1297–1324.
- [5] I. P. GAVRILYUK, W. HACKBUSCH, B. N. KHOROMSKIY, *Data-sparse approximation to a class of operator-valued functions*, Math. Comp., 74 (2005), pp. 681–708.
- [6] D. HENRY, *Geometric theory of semilinear parabolic equations*. Lecture Notes in Mathematics 840, Springer, Berlin, 1981.
- [7] M. HOCHBRUCK, C. LUBICH, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal. 34 (1997), pp. 1911–1925.
- [8] M. HOCHBRUCK, C. LUBICH, *Error analysis of Krylov methods in a nutshell*, SIAM J. Sci. Comput. 19 (1998), pp. 695–701.
- [9] M. HOCHBRUCK, A. OSTERMANN, *Exponential Runge-Kutta methods for parabolic problems*, Appl. Numer. Math. 53 (2005), pp. 323–339.
- [10] M. HOCHBRUCK, A. OSTERMANN, *Explicit exponential Runge-Kutta methods for semilinear parabolic problems*, SIAM J. Numer. Anal. 43 (2005), pp. 1069–1090.
- [11] A. K. KASSAM, L. N. TREFETHEN, *Fourth-order time-stepping for stiff PDEs*, SIAM J. Sci. Comput. 26 (2005), pp. 1214–1233.
- [12] M. LÓPEZ-FERNÁNDEZ, C. LUBICH, C. PALENCIA, A. SCHÄDLE, *Fast Runge-Kutta approximation of inhomogeneous parabolic equations*, Numer. Math., 102 (2005), pp. 277–291.

- [13] M. LÓPEZ-FERNÁNDEZ, C. PALENCIA, *On the numerical inversion of the Laplace transform of certain holomorphic mappings*, Appl. Numer. Math., 51 (2004), pp. 289–303.
- [14] M. LÓPEZ-FERNÁNDEZ, C. PALENCIA, A. SCHÄDLE, *A spectral order method for inverting sectorial Laplace transforms*, SIAM J. Numer. Anal., 44 (2006), pp. 1332–1350.
- [15] W. MCLEAN, V. THOMÉE, *Time discretization of an evolution equation via Laplace transforms*, IMA J. Numer. Anal., 24 (2004), pp. 439–463.
- [16] D. SHEEN, I. H. SLOAN, V. THOMÉE, *A parallel method for time discretization of parabolic equations based on Laplace transformation and quadrature*, Math. Comp., 69 (2000), pp. 177–195.
- [17] F. STENGER, *Approximations via Whittaker’s Cardinal Function*, J. Approx. Theory, 17 (1976), pp. 222–240.
- [18] F. STENGER, *Numerical methods based on Whittaker Cardinal, or sinc Functions*, SIAM Rev., 23 (1981), pp. 165–224.
- [19] A. TALBOT, *The accurate numerical inversion of Laplace transforms*, J. Inst. Math. Appl., 23 (1979), pp. 97–120.
- [20] J. A. C. WEIDEMAN, *Optimizing Talbot’s contours for the inversion of the Laplace transform*, SIAM J. Numer. Anal., 44 (2006), pp. 2342–2362.
- [21] J. A. C. WEIDEMAN, L. N. TREFETHEN, *Parabolic and hyperbolic contours for computing the Bromwich integral*, Math. Comput. 76 (2007), pp. 1341–1356.